به نام هستی بخش

" و سلام بر مهدی که انتظارش را نه فقط دل عاشق،
که ترنم هر باران بهاری و هر روزنهی امید میکشد..."

تمرین 8 (فصل 12 و 13)

موعد تحویل:

درس پایگاه داده ها، بهار 1403

---

1. به سوالات زیر پاسخ دهید.

الف) استفاده از متد Chunked-IO چگونه هزینه IO را کاهش میدهد؟ چه trade-off ای بین block-size و هزینه وجود دارد؟

این روش با انجام IO به صورت بلوکی, هزینه IO به صورت sequential را کاهش میدهد
trade-off هم به این شکل است که به ازای افزایش اندازه بلوک, fan-out مرج شدن کم شده و ممکن است باعث افزایش تعداد pass ها شده و هزینه بالا رود.

ب) کدام بخش از عملگر projection هزینه بر است؟ روش های آن را نام برده و توضیح کوتاه دهید.

بخش حذف کردن duplicate ها که در sql با کلید DISTINCT گفته میشود
از روش های آن میتوان به
sort: برا اساس شمای خواسته شده, رابطه را مرتب کرده و به این ترتیب تکراری ها کنار هم میوفتند و آنها را حذف میکنند
hash: آنها را در bucket هایی ذخیره میکنند و تکراری ها حتما در یک bucket می افتند و آنها را در حافظه اصلی لود کرده و حذف میکنند.

ج) آیا میتوان در یک پلن fully-piped-line از روش sort-merge-join استفاده کرد؟ توضیح دهید.

بله, اگر هر دو relation از قبل بر اساس کلید join مرتب شده باشند (مثلا اگر روی هر کدام یک clustered tree داشته باشیم) میتوان بدون مرتب سازی و ذخیره کردن آنها در temp فایل ها, join را انجام داد.

د) سناریویی مثال بزنید که هزینه external merge sort با internal merge sort یکی باشد. (N و B را پیدا کنید).

کافیست N و B را طوری پیدا کنیم که این دو رابطه به یک عدد برسند:

$$1+\lceil \log_{B-1}(\lceil N/B \rceil) \rceil = 1+\lceil \log_{B-1}(\lceil N/2B \rceil) \rceil$$

2. برای دسترسی به هر یک از رکورد های زیر از کدام شاخص hash یا b+ tree بر روی (a,b,c,d) میتوان استفاده کرد؟ دلیل انتخاب یا عدم انتخاب هر شاخص را بیان کنید.

a) a > 9 AND b < 8

از hash نمیتوان استفاده کرد چون range search داریم. ولی از درخت میتوان استفاده کرد چون یک query از prefix شاخص است البته باید درخت clustered باشد

b) a = 1 AND (b < 7 OR c = 2)

تنها میتوان بخش a = 1 را با شاخص انجام داد, زیرا عملگر OR را نمیتوان روی رکورد های داده ای با شاخص داشت و به اگر هر دو شاخص clustered باشند, معمولا hash بهتر است زیرا پیدا کردن اینکدس 1.2IO و برای درخت به اندازه ارتفاع آن که معمولا 2 تا 4 است هزینه دارد

c) a = 1 OR (b < 7 AND c = 2)

برای این query نمیتوان از شاخص استفاده کرد زیرا در خارج عملگر OR داریم, حتی داخل پرانتز هم اگر از شاخص استفاده کنیم, باز هم باید کل relation را scan کنیم و داخل حافظه این شروط را چک کنیم.

3. فرض کنید query روبه رو به شما داده شده : day<8/9/98 AND bid=6 AND sid=102
و شاخص های b+ tree روی day و hash روی (bid,sid) است.
الف) با تعیین کردن نوع clustered/unclustered بودن هر کدام, 3 سناریو طراحی کنید که در اولی شاخص hash بهتر باشد, در دومی شاخص b+ tree و در سومی scan کردن کل فایل ها بهتر باشد.
(فرض کنید کل relation دارای 100صفحه و 20000 رکورد است و reduction factor هر ترم 0.1 است)
سناریو اول: hash از نوع clustered و درخت unclustered باشد. پس هزینه hash به اندازه 100*0.1*0.1 یعنی 1 IO است ولی درخت 0.1 * 20000 = 2000 IO است

سناریو دوم:hash از نوع unclustered و درخت clustered باشد. پس هزینه hash به اندازه 20000*0.1*0.1 یعنی 200 IO است ولی درخت 0.1 * 100 = 10 IO است.

سناریو سوم: هر دو شاخص از نوع unclustered باشند. پس هزینه hash به اندازه 20000*0.1*0.1 یعنی 200 IO و هزینه درخت به اندازه 20000*0.1= 2000 IO است
ولی scan فایل به اندازه 100 IO یعنی به تعداد صفحه های relation است.

ب) query optimizer از کجا متوجه میشود که کدام سناریو پیش آمده و اطلاعات relation را از کجا دارد؟
از طریق catalog سیستم که اطلاعات relation ها و شاخص ها و آمار های آنها را دارد.

4. با توجه به شمای زیرو توضیحات آن به سوالات زیر پاسخ دهید:

Students (sid, sname, gpa): 300 Pages, 20 tuple/page

Enrollments (sid, cid, quantity): 600 pages, 150 tuple/page

Courses (cid, cname, classNo): 150 pages, 20 tuple/page

یک شاخص clustered tree روی sid در students

یک شاخص unclustered hash روی sid,cid بر روی enrollments

الف) کمترین هزینه join کردن دو رابطه students و enrollments را به ازای دو روش PONLJ و INLJ به دست بیاورید.

ب) اگر قرار باشد از روش external merge sort join استفاده کنیم, فقط برای بخش sort کردن, هزینه چقدر میشود؟ (تعداد صفحه های بافر را 5 در نظر بگیرید)

۴۰

الف) PONLJ: $M + M \times N$

$= ۳۰۰ + ۳۰۰ \times ۴۰۰ = ۱۲۰۳۰۰$ IO outer: students

INLJ: M_pages + M_tuples x (lookup time)

inner: students → $۴۰۰ + ۴۰۰ \times ۱۲۰ \times (1) = ۹۰۴۰۰$ IO

inner: enrollments → $۳۰۰ + ۳۰۰ \times ۲۰ \times (1 \times ۱۰) =$ ۶۰۳۰۰ IO

ب) به دلیل داشتن clustered tree روی sid در student

می‌دانیم که صفحه‌های آن از قبل مرتب شده‌اند پس:

sort students = ۳۰۰ IO (خواندن صفحات)

sort enrollments = $۲ \times ۴۰۰ \times (1 + \log_{۴}^{۱۲۰}) = ۳۴۰۰$ IO

→ هزینه sort = ۳۹۰۰ IO

5. این سوال را نیز با توجه شمای سوال قبل و شاخص های زیر:

شاخص های students:

unclustered tree on (gpa, sid)

clustered hash on sid

شاخص های courses:

unclustered hash on sid

unclustered tree on cname

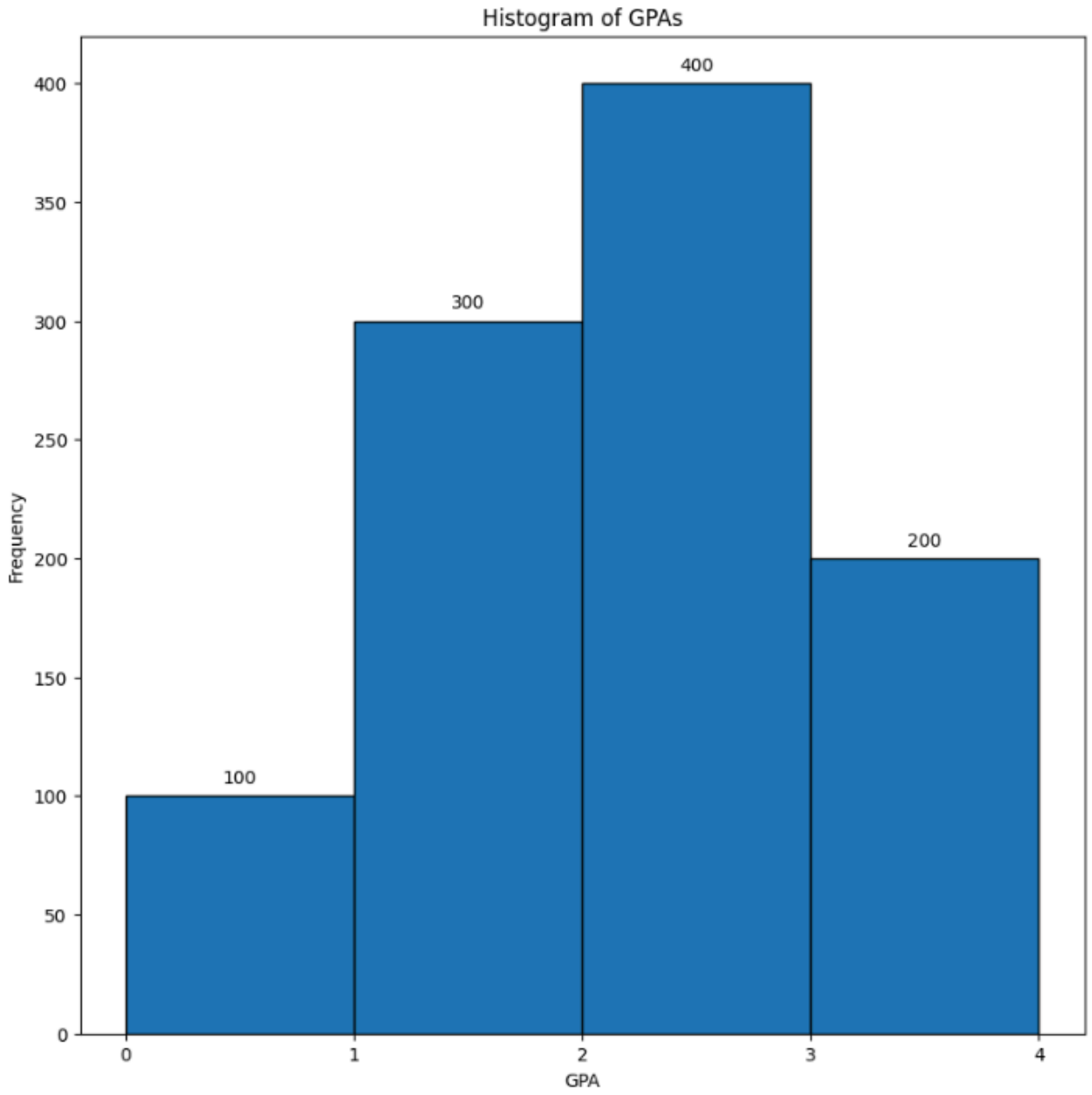
شاخص های enrollments:

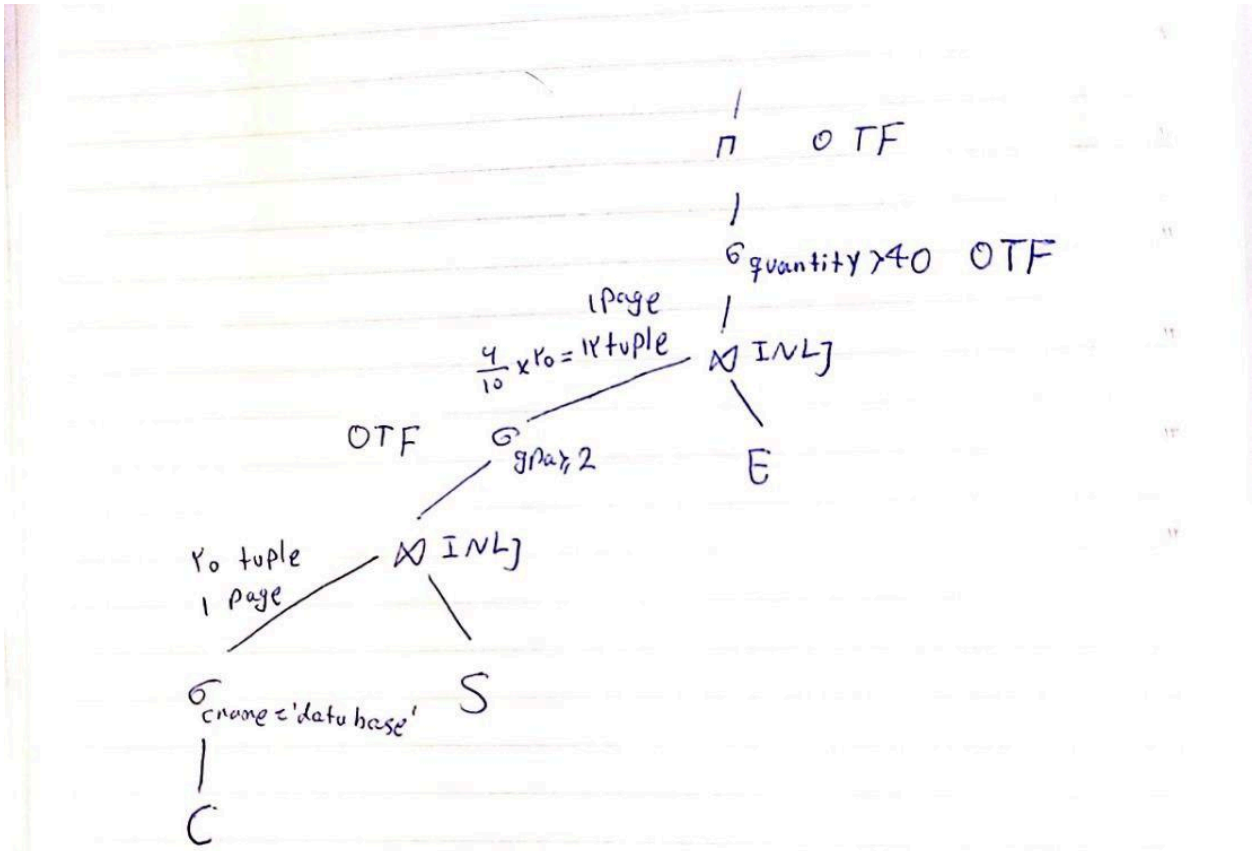
clustered on quantity

برای query زیر, پلن با کمترین هزینه را نشان داده و هزینه آن را به دست آورید:

```
Select S.sid, C.cid
From Students S, Enrollments E, Courses C
Where S.sid = E.sid AND C.cid = E.cid AND S.gpa >= 2 AND E.quantity > 40 AND C.cname = 'Database'
```

فرض کنید که توزیع quantity کلاس ها به صورت یکنواخت از 10 تا 60 است, برای درس دیتابیس 10 کورس برگزار شده و توزیع gpa دانشجویان به شکل زیر است:


Histogram of GPAs
Frequency
GPA
0
50
100
150
200
250
300
350
400
0
1
2
3
4
100
300
400
200



ارتفاع درخت ← تعداد رکوردها

Cost($\sigma_{cname='data\ base'}$) = ۲ + ۲۰ = ۲۲ IO

تعداد رکوردها

Cost(C ⋈ S : INLJ) = ۱ + ۲۰ x (۱ + ۱۰) = ۲۲۱ IO

hash

ارتفاع درخت ← تعداد رکورد

Cost(⋈ E : INLJ) = ۱ + ۱۲ x (۲ + ۱۰) = ۱۴۱ IO

Cost(Total) = ۳۹۲ IO

6. اگر بخواهیم فایلی با 10000 صفحه را با 8 صفحه بافر و با روش internal merge sort مرتب کنیم:

الف) چه تعداد pass برای مرتب سازی نیاز است؟ هزینه IO چقدر خواهد شد؟

ب) تعداد و اندازه run های تولید شده در هر pass را مشخص کنید

ج) فرض کنید میخواهیم هزینه IO را با روش chunked-IO کاهش دهیم, سایز بلاک را طوری به دست بیاورید که تعداد pass ها ثابت بماند.

4.

الف) تعداد pass ها = $1 + \lceil \log_{B-1} \lceil \frac{N}{2B} \rceil \rceil = 1 + \lceil \log_7 \overbrace{\lceil 4500 \rceil}^{4500} \rceil = 5$

هزینه = $2N \times$ (تعداد pass) $= 2 \times 10^6 \times 4 = 8000000$ IO

ب) Pass 0: تعداد run ها $= \lceil \frac{10^6}{14} \rceil = 4500$ ← اندازه هر run

Pass 1: تعداد run ها $= \lceil \frac{4500}{7} \rceil = 643$ ← اندازه هر run $= 14 \times 7 = 112$

و یک run با اندازه 32

Pass 2: تعداد run ها $= \lceil \frac{700}{7} \rceil = 92$ ← اندازه هر run $= 7 \times 7 \times 7 = 343$

و یک run با اندازه 84

Pass 3: تعداد run ها $= \lceil \frac{13}{7} \rceil = 2$ ← اندازه هر run $= 7 \times 7 \times 7 \times 7 = 2401$

و یک run با اندازه 1204

Pass 4: یک run با اندازه 10000